ZBLD.C20 series 485 MODBUS (RTU) protocol

1. Data frame format

<table>
<tr><td colspan="5">Modbus -- RTU mode</td></tr>
<tr><td>Address code</td><td>Function code</td><td colspan="2">data</td><td colspan="2">Check code</td></tr>
<tr><td>ADDR</td><td>CMD</td><td colspan="2">DATA</td><td>CRC16_L</td><td>CRC16_H</td></tr>
<tr><td>1 Byte</td><td>1 Byte</td><td colspan="2">N Bytes</td><td>1 Byte</td><td>1 Byte</td></tr>
<tr><td colspan="6">Baud rate: 19200 (default) start bit: 1bit data bit: 8bit check bit: no check stop bit: 1bit</td></tr>
<tr><td colspan="6">In Modbus RTU mode, when the time interval between each two characters is more than 1.5 times, the data is invalid; if the time interval between two characters is more than 3.5 times, it is considered that one frame of data has been transmitted according to regulations, and a new frame of data can be transmitted.</td></tr>
</table>

a) Address code: 1 byte

0x00 (0) --- address of broadcast, no reply from slave during broadcast, applicable to multiple group control

0x01 (1) - 0xf7 (247) --- slave address

0xf8 (248) - 0xff (255) --- system reserved, do not use

b) Function code: 1 byte

0x03 (3) --- read register operation

0x06 (6) --- write single register operation

0x10 (16) --- write multiple register operations

c) Data: n byte

Different instructions have different data formats!

d) Check code: 2 bytes

CRC16 verification object: address code + function code + data

CRC16 verification algorithm: MODBUS (x16 + x15 + x2 + 1)

1) 0x03 read register operation

0x03 (read register) data frame format

Host request			
Frame data	*Data length*	*Data content*	*Explain*
address	1 Byte	0x01-0xf7	
Function code	1 Byte	0x03	
Register address	2 Byte	0x2000-0x30ff	High byte + low byte
Number of registers	2 Byte	0x0001	High byte + low byte
Check code	2 Byte	CRC16L+CRC16H	Low byte + high byte
Respond after the slave receives the correct one			
Frame data	*Data length*	*Data content*	*Explain*
address	1 Byte	0x01-0xf7	
Function code	1 Byte	0x03	
Bytes	1 Byte	0x02	Total bytes in data area
data	2 Byte	Read data content	High byte + low byte
Check code	2 Byte	CRC16L+CRC16H	Low byte + high byte
Reply after receiving exception from slave			
Frame data	*Data length*	*Data content*	*Explain*
address	1 Byte	0x01-0xf7	
Function code	1 Byte	0x83	Msb=1
Exception code	1 Byte	0x00-0xff	See 4) exception code for details
Check code	2 Byte	CRC16L+CRC16H	Low byte + high byte

Note: all register addresses are 2byte, with high byte first and low byte second;

All register data bits are 2byte, with high byte first and low byte second;

This operation is only available for w / R or w / R * or R type registers;

0x03 (read holding register) example

a) Read 0x0001 register data of slave 0x01

Host	address	Function code	Register address		Number of registers		Check code	
			High byte	Low byte	High byte	Low byte	Low position	High position
	0x01	0x03	0x20	0x01	0x00	0X01	CRCL	CRCH
Slave machine	address	Function code	Bytes		data		Check code	
					High byte	Low byte		
	0x01	0x03	0x02		0x00	0X01	CRCL	CRCH

b) Reading 0x0001 register data of slave 0x01 is abnormal

Host	address	Function code	Register address		Number of registers		Check code	
			High byte	Low byte	High byte	Low byte	Low position	High position
	0x01	0x03	0x20	0x01	0x00	0X01	CRCL	CRCH
Slave machine	address	Function code	Exception code				Check code	
	0x01	0x83	See 4) exception code for details				CRCL	CRCH

2）0x06 write single register operation

0x06 (write single register) data frame format

Host request			
Frame data	*Data length*	*Data content*	*Explain*
address	1 Byte	0x01-0xf7	
Function code	1 Byte	0x06	
Register address	2 Byte	0x2000-0x30ff	High byte + low byte
data	2 Byte	Write data content	High byte + low byte
Check code	2 Byte	CRC16L+CRC16H	Low byte + high byte
Respond after the slave receives the correct one			
Frame data	*Data length*	*Data content*	*Explain*
address	1 Byte	0x01-0xf7	
Function code	1 Byte	0x06	
Register address	2 Byte	0x2000-0x30ff	High byte + low byte
Data bytes	2 Byte	Write data content	High byte + low byte
Check code	2 Byte	CRC16L+CRC16H	Low byte + high byte
Reply after receiving exception from slave			
Frame data	*Data length*	*Data content*	*Explain*
address	1 Byte	0x01-0xf7	
Function code	1 Byte	0x86	Msb=1
Exception code	1 Byte	0x00-0xff	See 4) exception code for details
Check code	2 Byte	CRC16L+CRC16H	Low byte + high byte

Note: all register addresses are 2byte, with high byte first and low byte second;

All register data bits are 2byte, with high byte first and low byte second;

This operation is only available for w / R or w / R * type registers;

0x06 (write single register) example

a) Write 0x2001 register of slave No. 0x01 to write data

Host	address	Function code	Register address		data		Check code	
			High byte	Low byte	High byte	Low byte	Low position	High position
	0x01	0x06	0x20	0x01	0x00	0X01	CRCL	CRCH
Slave machine	address	Function code	Register address		data		Check code	
			High byte	Low byte	High byte	Low byte		
	0x01	0x06	0x20	0x01	0x00	0X01	CRCL	CRCH

b) Write 0x0001 register data exception of slave 0x01

Host	address	Function code	Register address		data		Check code	
			High byte	Low byte	High byte	Low byte	Low position	High position
	0x01	0x06	0x20	0x01	0x00	0X01	CRCL	CRCH
Slave machine	address	Function code	Exception code				Check code	
	0x01	0x86	See 4) exception code				CRCL	CRCH

3) 0x10 write multiple register operation

0x10 (write multiple registers) data frame format

Host request			
Frame data	*Data length*	*Data content*	*Explain*
address	1 Byte	0x00-0xf7	
Function code	1 Byte	0x10	
Start register address	2 Byte	0x2000-0x30ff	High byte + low byte
Number of registers	2 Byte	N	High byte + low byte
Total data bytes	1 Byte	2*N	
data	2*N Byte		High byte + low byte
Check code	2 Byte	CRC16L+CRC16H	Low byte + high byte
Respond after the slave receives the correct one			
Frame data	*Data length*	*Data content*	*Explain*
address	1 Byte	0x01-0xf7	
Function code	1 Byte	0x10	
Start register address	2 Byte	0x2000-0x30ff	High byte + low byte
Number of registers	2 Byte	N	High byte + low byte
Check code	2 Byte	CRC16L+CRC16H	Low byte + high byte
Reply after receiving exception from slave			
Frame data	*Data length*	*Data content*	*Explain*
address	1 Byte	0x01-0xf7	
Function code	1 Byte	0x90	Msb=1
Exception code	1 Byte	0x00-0xff	See 4) exception code for details

Check code	2 Byte	CRC16L+CRC16H	Low byte + high byte

Note: all register addresses are 2byte, with high byte first and low byte second;

All register data bits are 2byte, with high byte first and low byte second;

This operation is only available for w / R or w / R * type registers;

0x10 (write multiple registers) example

Write 0x2000, 0x2001 register of slave 0x01 to write data

Host	address	Function code	Start register address		Number of registers		Total data bytes	
			High byte	Low byte	High byte	Low byte		
	0x01	0x10	0x20	0x00	0x00	0x02	0x04	
	Register 1 data		Register 2 data		Check code			
	High byte	Low byte	High byte	Low byte				
	Data 1		Data 2		CRCL	CRCH		
Normal return of slave	address	Function code	Start register address		Number of registers		Check code	
			High byte	Low byte	High byte	Low byte		
	0x01	0x10	0x20	0x00	0x00	0x02	CRCL	CRCH
Slave abnormal return	address	Function code	Exception code				Check code	
	0x01	0x90	See 4) exception code for details				CRCL	CRCH

4) exception code

Modbus exception code		
Exception code	*Meaning*	*Explain*
0x01	Illegal order	The slave does not support this command or processes the request in an error state;
0x02	Illegal data address	The request data address of the upper computer is out of range;
0x03	Illegal data value	The received data contains values that are not allowed;
0x04	operation failed	In parameter writing operation, the parameter is set to invalid setting, for example, write enable is not enabled when writing function code (0x200e);
0x05	Password error	The password written in the password validation address is incorrect;
0x06	Data frame error	The length of data frame is incorrect or the CRC check bit of RTU format is incorrect;
0x07	Parameter is read-only	When the upper computer writes, the changed parameters are read-only;
0x08	Parameter operation cannot be changed	The parameters changed when the upper computer writes are those that cannot be changed during operation;

2. Register list

Function description	Address definition	Data meaning	R/W characteristics
Communication control command	2000H	0001h: forward running	W/R
		0002h: reverse operation	
		0003h: forward turning inching	
		0004h: reverse jog	
		0005h: shutdown	
		0006h: free stop (emergency stop)	
		0007h: fault reset	
		0008h: inching stop	
Communication set point address	2001H	Communication set speed (0 ~ 3000 (unit: 1rpm))	W/R
	2002H	Motor pole pairs (1-20)	W/R*
	2003H	Acceleration time (1-6000 (unit: 0.1s))	W/R
	2004H	Deceleration time (1-6000 (unit: 0.1s))	W/R
	2005H	Control mode selection (0 ~ 3): 0:test mode 1:Ring opening 2:Closed loop 3:dial switch setting (SW1 valid), model related	W/R*
	2006H	Operation instruction selection (0-4): 0: keyboard operation command channel 1: terminal operation command channel 2: communication operation command channel 3: dial switch setting (SW3 valid), model related	W/R
	2007H	Speed setting selection (0-5): 0: keyboard number setting 1: analog quantity Ai1 setting (knob potentiometer) 2: analog quantity AI2 setting (external voltage) 3: Modbus communication setting 4: multi segment speed setting	W/R

		5: dial switch setting (SW2, SW3 valid), model related 6: simple PLC setting	
	2008H	Local communication address 1 ~ 247, 0 is broadcast address	W/R*
	2009H	Communication baud rate setting (0 ~ 6): 0:1200BPS 1:2400BPS 2:4800BPS 3:9600BPS 4:19200bps (default) 5:38400BPS 6:57600BPS Note: only some models can be set	W/R*
	200AH	Virtual input terminal command, range: 0x000 ~ 0x1ff	W/R
	200BH	Virtual output terminal command, range: 0x00 ~ 0x0f	W/R
	200EH	Write operation enable of communication function code: (for group f00-f10 function code) 0: function code cannot be written during communication (default) 1: function code can be written during communication	W/R
	200FH	Function code restore default value: 0No operation. 1: restore default parameters	W/R*
Drive status word 1	2100H	0001h: forward running	R
		0002h: in reverse operation	
		0003h: drive in shutdown	
		0004h: drive in fault	
		0005h: drive off	
		0006h: electronic brake status	

Drive status word 2	2101H	Bit0: = 0: bus voltage not established =1: Bus voltage establishment Bit4: = 0: not overloaded =1: overload Bit5~ Bit6： =00: keyboard control =01: terminal control =10: Communication control	R
Drive fault code	2102H	See fault type description	R
Drive ID	2103H	C20 series -----0x0020	R
Set frequency	3000H	0 ~ Fmax (unit: 0.01Hz)	R
output frequency	3001H	0 ~ Fmax (unit: 0.01Hz)	R
Given frequency of slope	3002H	0 ~ Fmax (unit: 0.01Hz)	R
output voltage	3003H	0.0 ~ 2000.0v (unit: 0.1V)	R
Output current	3004H	0.0 ~ 300.00a (unit: 0.01A)	R
Set speed	3005H	0 ~ 3000 (unit: 1rpm)	R
Motor output speed	3006H	0 ~ 3000 (unit: 1rpm)	R
Motor output power	3007H	0~2200W	R
DC Bus Voltage	3008H	0.0 ~ 2000.0v (unit: 0.1V)	R
Holzer value	3009H	0~7	R
Software version number	300AH	1.00~99.00	R
Current fault type	300BH	See fault type description	R
Inverter temperature	300CH	-20.0℃ ~ 120 ℃ (supported by some hardware)	R
Input terminal status	300DH	000~1FF	R
Output terminal status	300EH	00~0F	R
Input voltage of analog quantity 1	300FH	0.00 ~ 10.00v (unit: 0.01V)	R
Input voltage of analog	3010H	0.00 ~ 10.00v (unit: 0.01V)	R

quantity 2			
Analog 3 input voltage	3011H	0.00 ~ 10.00v (unit: 0.01V)	R

Parameter Property Description:

attribute	Explain
W/R	Indicates that the set value of the parameter can be read and written when the driver is in any state;
W/R*	Indicates that the set value of the parameter can be written when the driver is in the shutdown state, and can be read in any state;
R	Indicates that the parameter is read-only and cannot be written;

3. Common command examples

Slave address 0x01, baud rate 19200, n-8-1

operation	Data frame	Explain
Forward turn operation	Send data: 01 06 20 00 01 43 CA Response data: 01 06 20 00 01 43 CA	Address 2000h write 0x01
Reverse operation	Send data: 01 06 20 00 02 03 CB Response data: 01 06 20 00 02 03 CB	Address 2000h write 0x02
Shutdown	Send data: 01 06 20 00 05 42 09 Response data: 01 06 20 00 05 42 09	Address 2000h write 0x05
Fault reset	Sending data: 01 06 20 00 07 C3 C8 Response data: 01 06 20 00 07 C3 C8	Address 2000h write 0x07
Set speed	Sending data: 01 06 20 01 0b B8 D4 88 Response data: 01 06 20 01 0b B8 D4 88	Address 2001h write 0x0bb8 Set speed 3000rpm
Enable function code writable	Send data: 01 06 20 0e 00 E3 C9 Response data: 01 06 20 0e 00 E3 C9	Address 200fh write 0x01
Read function code	Sending data: 01 03 00 0A 00 01 A4 08 Response data: 01 03 02 13 88 B5 12	The value of reading function code f00.10 is 50.00Hz (1388h)
Write function code	Send data: 01 06 00 0A 09 C4 AE 0b Response data: 01 06 00 0A 09 C4 AE 0b	The value of f00.10 is 25.00hz (09c4h)
Read status word 1	Send data: 01 03 21 00 01 8e 36 Response data: 01 03 02 00 05 78 47	Read address 2100h
Read status word 2	Send data: 01 03 21 01 00 01 DF F6 Response data: 01 03 02 00 41 78 74	Read address 2101h
Read trouble codes	Send data: 01 03 21 02 00 01 2F F6 Response data: 01 03 02 00 0A 38 43	Read address 2102h Fault code: 0x0a undervoltage fault

operation	Data frame	Explain
Fail to read	Send data: 01 03 00 32 00 01 25 C5 Answer data: 01 83 02 C0 F1	Failed to read function code f00.50 Address error: 0x02
Write failure	Send data: 01 06 00 00 05 49 C9 Response data: 01 86 03 02 61	Write function code f00.00 to 5 Data error: 0x03